

Segmentación de Imágenes SAR usando Filtros Stack y Curvas B-spline

María E. Buemi, Juliana Gambini, Marta Mejail, Julio Jacobo Berllés

Facultad de Ciencias Exactas y Naturales, UBA, Buenos Aires, Argentina
Ciudad Universitaria, Pabellón I.

(C1428EGA) Buenos Aires. República Argentina.

Tel.: +54-11-4576-3390/96 int 701/702

{mebuemi; jgambini; marta; jacoboj}@dc.uba.ar

Resumen Las imágenes generadas con sistemas de iluminación coherente, como las de Radar de Apertura Sintética (SAR) son muy difíciles de segmentar porque poseen ruido *speckle*. El ruido *speckle* es especialmente complejo de remover, debido a su naturaleza no aditiva. Estas imágenes pueden tener zonas homogéneas, heterogéneas o muy heterogéneas, correspondientes a zonas de pastura, bosque o urbanas, respectivamente. La extracción de los bordes entre las diferentes regiones es un tema importante en análisis de imágenes y se utiliza en diversas aplicaciones. En este trabajo se presenta la aplicación de un filtro adaptativo no lineal, denominado *stack filter*, que tiene el objetivo de remover el ruido *speckle* y mejorar la imagen para facilitar la búsqueda de características. Está basado en el entrenamiento con dos imágenes, una que se supone ideal (sin ruido) y otra imagen con un ruido *speckle*. Con el fin de hallar bordes de diferentes regiones en la imagen, se aplica un algoritmo de extracción de contornos basado en curvas B-spline y una distribución estadística adecuada. El algoritmo propuesto se aplica a imágenes SAR reales y se obtienen muy buenos resultados.

1. Introducción

Los filtros *stack* son un caso especial de filtros no lineales. Tienen buena performance para filtrado de imágenes con distintos tipos de ruido preservando bordes y detalles. Estos filtros consisten en una descomposición por umbrales de la señal de entrada, obteniéndose una señal binaria para cada uno de ellos, y en una aplicación de un filtro a cada una de esas señales por medio de una ventana deslizante.

Este filtro se genera por medio de un algoritmo adaptativo, de manera que cumpla con la propiedad de *stacking*. El método propuesto en este trabajo está basado en el algoritmo de Yoo *et al.* [9], adaptado a imágenes SAR.

Por otro lado las imágenes SAR poseen ruido *speckle* lo cual hace que sean muy difíciles de interpretar por su naturaleza multiplicativa. El ruido *speckle* es no-aditivo, no gaussiano y es lo que le da a la imagen esa apariencia granulada. Los métodos clásicos de procesamiento de

imágenes no pueden aplicarse en imágenes SAR porque utilizan la información local presente en un pixel. El ruido *speckle* es la razón por la cual debe utilizarse la información sobre regiones. El análisis de este tipo de imágenes ha sido tratado en la literatura con diversos métodos estadísticos (ver por ejemplo, [5], [7], [4], [3]). En este trabajo se aplica el filtro para segmentar imágenes con ruido *speckle*. Se remueve el ruido para luego encontrar contornos de regiones utilizando un algoritmo de curvas B-spline deformables basado en regiones, combinado con un modelo estadístico.

Este artículo está organizado de la siguiente manera: En la sección 2 se explican los principios de los filtros stack. En la sección 3 se presenta el algoritmo desarrollado para encontrar fronteras de regiones. En la sección 4 se introduce la justificación de las distribuciones estadísticas utilizadas en la segmentación. En la sección 5 se muestran los resultados obtenidos, y finalmente, en la sección 6 se presentan las conclusiones.

2. Los Filtros Stack: Definiciones y Diseño

En esta sección se presenta una breve síntesis de filtros *stack*. Para obtener una información más detallada sobre este tema, consultar [1]. En primer lugar se exponen las definiciones necesarias para explicar este tipo de filtros. Luego se introduce una síntesis de algoritmos adaptativos.

Definición 1

El operador de umbral $T^m : \{0, 1, \dots, M\} \rightarrow \{0, 1\}$ está dado por

$$T^m(x) = \begin{cases} 1 & \text{si } x \geq m \\ 0 & \text{si } x < m \end{cases}, \quad (1)$$

con $M, m \in \mathbb{Z}$ y donde m es el valor del umbral. Denotaremos

$$X^m = T^m(x). \quad (2)$$

De acuerdo a esta definición, el valor de un entero $x \in \{0, 1, \dots, M\}$ puede ser reconstruido realizando la sumatoria de sus valores umbrados entre 0 y M . La fórmula correspondiente a esta operación es

$$x = \sum_{m=1}^M X^m. \quad (3)$$

La Figura 1 muestra un diagrama de descomposición de umbrales para una señal unidimensional.

El operador de umbral puede extenderse a señales bi-dimensionales.

Definición 2

Sean $X = (x_0, \dots, x_{n-1})$ e $Y = (y_0, \dots, y_{n-1})$ vectores binarios de longitud n . Se define una relación \leq dada por

$$X \leq Y \quad \text{si y sólo si} \quad \forall i, x_i \leq y_i. \quad (4)$$

Esta relación es reflexiva, antisimétrica y transitiva, constituyendo así un ordenamiento parcial sobre el conjunto de vectores binarios de longitud fija.

$$\begin{aligned}
X &= [2, 1, 4, 5, 3, 2, 4, 3] \\
X^1 &= [1, 1, 1, 1, 1, 1, 1, 1] \\
X^2 &= [1, 0, 1, 1, 1, 1, 1, 1] \\
X^3 &= [0, 0, 1, 1, 1, 0, 1, 1] \\
X^4 &= [0, 0, 1, 1, 0, 0, 1, 0] \\
X^5 &= [0, 0, 0, 1, 0, 0, 0, 0]
\end{aligned}$$

Figura 1: Ejemplo de descomposición por umbrales

Definición 3

Una función booleana $f : \{0, 1\}^n \rightarrow \{0, 1\}$, donde n es la longitud de los vectores de entrada, posee la **propiedad de stacking** sii se cumple que

$$\forall X, Y \in \{0, 1\}^n, \quad X \leq Y \Rightarrow f(X) \leq f(Y). \quad (5)$$

Definición 4

Diremos que f es una función booleana positiva si y sólo si puede ser escrita mediante una expresión que contenga solamente variables de entrada no complementadas. O sea,

$$f(x_1, x_2, \dots, x_n) = \bigvee_{i=1}^K \bigwedge_{j \in P_i} x_j, \quad (6)$$

donde N es el número de argumentos de la función, K es la cantidad de términos de la expresión y los P_i son subconjuntos del intervalo $\{1, \dots, N\}$. \bigvee y \bigwedge son los operadores booleanos AND y OR. Se puede probar que este tipo de funciones posee la propiedad de stacking.

Si la función f utilizada para realizar el filtrado de una imagen X cumple con la propiedad de stacking, entonces de (4) y (5) se deduce que, para dos imágenes binarias X^i y X^j , obtenidas de X como resultado de la aplicación de los umbrales T^i y T^j respectivamente, resulta

$$i \geq j \Rightarrow X^i \leq X^j \Rightarrow f(X^i) \leq f(X^j). \quad (7)$$

Un filtro *stack* está definido por la función $S_f : \{0, \dots, M\}^n \rightarrow \{0, \dots, M\}$, correspondiente a la función Booleana positiva $f(x_1, x_2, \dots, x_n)$ expresada en la forma dada por (6). La función S_f puede ser expresada mediante

$$S_f(X) = \sum_{m=1}^M f(T^m(X)) \quad (8)$$

En el diseño de los filtros stack, la generación de la función booleana correspondiente tiene especial importancia. Varios autores han propuesto diversos métodos para esta tarea. Se destacan los algoritmos adaptativos (basados en entrenamiento) pero anteriormente a estos se habían desarrollado algoritmos para el diseño de *stack filters* que requerían conocimiento estadístico detallado (estadísticas de cruce de nivel). En todos los casos el objetivo es conseguir una función

booleana positiva que cumpla la propiedad de stacking.

La función booleana positiva se obtiene por medio de un algoritmo que realiza cambios de estado hasta alcanzar su estado final que cumpla con las propiedades mencionadas anteriormente. Resulta, entonces, un algoritmo iterativo que genera una función booleana positiva óptima para el filtrado de un tipo de imágenes dado. La actualización, en cada paso se hace en el sentido de la minimización del error entre la imagen filtrada y la imagen deseada (imagen ideal).

Un algoritmo adaptativo de referencia puede verse en [9].

3. Detección de Bordes

En esta sección se describe un algoritmo desarrollado para detección de bordes, usando contornos B-spline deformables y una distribución estadística que describa los datos. En la sección 3.1 se presenta un breve resumen de curvas B-spline, en la sección 3.2 se muestra el algoritmo de detección de bordes.

3.1. Representación B-spline para curvas

La representación B-spline para curvas es muy utilizada en procesamiento de imágenes, computación gráfica y visión, porque provee una herramienta muy eficaz en la descripción de curvas. En este trabajo se utiliza la representación B-spline para curvas con el propósito de describir el contorno de los objetos. Esta formulación tiene varias ventajas:

1. La curva queda representada por sus puntos de control, lo cual reduce la cantidad de parámetros y por lo tanto el costo computacional. Esto implica, por ejemplo, que si se quiere aplicar una transformación afín (rotación, traslación, escalamiento) a la curva, es suficiente aplicarla a sus puntos de control.
2. Puede elegirse el orden de la curva de acuerdo a la suavidad necesaria para el contorno del objeto de interés.
3. Dada su característica de control local, permite modificar la curva localmente por medio de sus puntos de control.
4. Tiene la propiedad de la cápsula convexa, esto impide divergencias en la curva y gráficos fuera del viewport.

En esta sección se presenta un breve resumen de la representación B-spline de contornos y de la interpolación de puntos por medio de B-spline utilizado para el ajuste; para más detalles sobre este tema consultar [2, 8].

Sea $\{Q_0, \dots, Q_{N_B-1}\}$ un conjunto de puntos de control, donde $Q_n = (x_n, y_n)^t \in R^2$, $0 \leq n \leq N_B - 1$, y sea $\{s_0 < s_1 < s_2 < \dots < s_{L-1}\} \subset R$ un conjunto de L nodos. Una curva B-spline de orden d se define como una suma ponderada de N_B funciones polinomiales $B_{n,d}(s)$ de grado $d - 1$, dentro del intervalo $[s_i, s_{i+1}]$ con $0 \leq i \leq L - 1$. La curva B-spline se contruye como $r(s) = (x(s), y(s))^t$, $0 \leq s \leq L - 1$, siendo

$$r(s) = \sum_{n=0}^{N_B-1} B_{n,d}(s)Q_n, \quad (9)$$

y

$$x(s) = B^t(s).Q^x \quad (10)$$

$$y(s) = B^t(s).Q^y \quad (11)$$

donde el vector de funciones base $B(s)$ de N_B componentes está dado por $B(s) = (B_{0,d}(s), \dots, B_{N_B-1,d}(s))^t$. Los vectores de pesos Q^x y Q^y son las primeras y segundas componentes de Q_n , respectivamente.

Las curvas usadas en este trabajo para extracción de contornos son cerradas, de orden $d = 3$ ó $d = 4$, especificadas por funciones B-spline periódicas.

Para resolver el problema de encontrar los N_B puntos de control de una curva B-spline que ajuste un contorno, utilizamos el siguiente esquema desarrollado en [8].

Dado un conjunto de k puntos en el plano de la imagen $\{D_0, D_1, \dots, D_{k-1}\}$, donde $D_i = (x_i, y_i)^t$, $i = 0, \dots, k-1$, buscamos la curva B-spline de N_B puntos de control que mejor los ajusta. Entonces, por las ecuaciones 10 y 11, las componentes D_i deben satisfacer

$$x_i = B^t(t_i)Q^x, \quad (12)$$

$$y_i = B^t(t_i)Q^y, \quad (13)$$

para ciertos valores de t_i , donde $i = 0, \dots, k-1$ y $N_B \leq k$.

Este sistema lineal puede ser escrito en forma matricial como $D = K(Q^x \ Q^y)$, donde los $k \times N_B$ elementos de la matriz K están dados por $K_{ij} = B_{j,d}(t_i)$, con $i = 0, \dots, k-1$, $j = 0, \dots, N_B-1$, y $D = (D_0, D_1, \dots, D_{k-1})^t$.

En el caso más general es $N_B < k$ y por lo tanto K no es una matriz cuadrada. En este caso se utiliza la matriz pseudo-inversa para encontrar una solución aproximada del sistema, de donde resulta

$$(Q^x \ Q^y) = K^+ D$$

El conjunto de valores de los parámetros $\{t_0, \dots, t_{k-1}\}$ está dado por

$$t_0 = 0, \ ; t_\ell = \frac{\sum_{i=1}^{\ell} \|D_i - D_{i-1}\|}{\sum_{i=1}^{k-1} \|D_i - D_{i-1}\|}, \ell \geq 1.$$

3.2. Algoritmo de Rectas Radiales para Detección de Puntos de Borde

Sea $E : R^2 \rightarrow R$ una escena compuesta por el fondo B y un conjunto de regiones $\{R_1, R_2, \dots, R_k\}$ con sus respectivas fronteras $\{\partial R_1, \dots, \partial R_k\}$. Deseamos encontrar curvas C_j que ajusten al borde cada región ∂R_j , $j = 1, \dots, k$, en la imagen. Cada una de estas regiones y el fondo son consideradas como campos aleatorios de variables aleatorias independientes e idénticamente distribuidas con distribución \mathcal{D} y caracterizadas por los valores de sus parámetros estadísticos $\lambda_1, \dots, \lambda_D$.

En el primer paso se encuentra una aproximación de estas regiones, las cuales son consideradas de interés y usadas para inicializar métodos de aplicación posterior. Definimos un área inicial de búsqueda por medio de polígonos cuyos vértices son puntos de control que generan

una curva B-spline. Una vez que las zonas de búsqueda inicial son determinadas se calcula su centroide.

Si un punto pertenece al borde del objeto entonces en un entorno de ese punto existe un cambio en los parámetros estadísticos de la muestra y por lo tanto es un punto de transición. Se consideran N segmentos sobre la imagen, $s^{(i)}$, $i \in \{1, \dots, N\}$, de la forma $s^{(i)} = \overline{CP_i}$ donde C es el centroide de la región inicial, el extremo P_i es un punto fuera de la región y $\theta = \text{ang}(s^{(i)}, s^{(i+1)})$ $\forall i$ es el ángulo entre dos segmentos sucesivos.

El segmento $s^{(i)}$ es un arreglo cuyos elementos provienen de la discretización de la línea sobre la imagen.

$$s^{(i)} = \left(z_1^{(i)}, \dots, z_m^{(i)} \right), \quad (14)$$

Dado un segmento $s^{(i)}$ consideramos la siguiente partición

$$\begin{aligned} Z_k^{(i)} &\sim \mathcal{D}(\lambda_{1r}, \dots, \lambda_{Nr}), \quad k = 1, \dots, j \\ Z_k^{(i)} &\sim \mathcal{D}(\lambda_{1b}, \dots, \lambda_{Nb}), \quad k = j+1, \dots, m \end{aligned}$$

donde para cada k , $z_k^{(i)}$ es la realización de la variable aleatoria $Z_k^{(i)}$. Los parámetros $(\lambda_{1r}, \dots, \lambda_{Nr})$ and $(\lambda_{1b}, \dots, \lambda_{Nb})$ caracterizan a la región y al fondo, respectivamente. Para encontrar los puntos de transición sobre cada segmento, consideramos una función ℓ para cada segmento $s^{(i)}$, donde ℓ es una función de verosimilitud de la muestra, dada por:

$$\ell(z, \lambda_{1r}, \dots, \lambda_{Nr}, \lambda_{1b}, \dots, \lambda_{Nb}) = \prod_{i=0}^j \text{Pr}(z_i; \lambda_{1r}, \dots, \lambda_{Nr}) \prod_{i=j+1}^m \text{Pr}(z_i; \lambda_{1b}, \dots, \lambda_{Nb})$$

Para encontrar el punto de transición, maximizamos $L = \ln \ell$:

$$L = \sum_{i=0}^j \ln(f_{\mathcal{D}}(z_i; \lambda_{1r}, \dots, \lambda_{Nr})) + \sum_{i=j+1}^m \ln(f_{\mathcal{D}}(z_i; \lambda_{1b}, \dots, \lambda_{Nb}))$$

donde $f_{\mathcal{D}}$ es la función de densidad de la distribución \mathcal{D} . Luego, la estimación del índice \hat{j} que corresponde al punto de borde b_i , sobre el segmento $s^{(i)}$ está dado por:

$$\hat{j} = \arg \max_j L \quad (15)$$

Estimamos los parámetros estadísticos procesando los pixels dentro de una ventana rectangular. La dirección del eje mayor de la ventana coincide con el segmento procesado. La ventana incluye una muestra de la región y otra del fondo. Estas muestras son utilizadas para estimar los parámetros $(\lambda_{1r}, \dots, \lambda_{Nr})$ y $(\lambda_{1b}, \dots, \lambda_{Nb})$ respectivamente.

En el caso de que la ventana rectangular esté totalmente incluida dentro de la región o totalmente incluida en el fondo, los parámetros que se estimen no corresponderán a diferentes regiones y el proceso de encontrar los puntos de borde $\{b_1 \dots b_N\}$ debe ser reconsiderado buscando otra posición para el centro de la ventana.

El Algoritmo 1 muestra el resumen del proceso de detección de bordes.

Algorithm 1 Algoritmo de Detección de Bordes

- 1: Determinar la región de interés por medio de una curva b-spline.
 - 2: Determinar una serie de segmentos radiales sobre la imagen.
 - 3: **for** cada segmento **do**
 - 4: Estimar los parámetros estadísticos de la distribución.
 - 5: Encontrar los puntos de borde b_i resolviendo la Ec. 15.
 - 6: **end for**
 - 7: Construir la curva b-spline interpolante para el conjunto de puntos $\{b_1, \dots, b_N\}$.
-

4. Detección de Bordes en Imágenes Filtradas

El algoritmo de detección de bordes descrito en la sección 3 utiliza un modelo estadístico para caracterizar las distintas regiones de la imagen basados en el grado de homogeneidad de sus datos. Por ejemplo, en el caso de las imágenes SAR monopolarizadas sin filtrar, se puede trabajar con la distribución \mathcal{G} , como se describe en los artículos [3] y [4]. En el trabajo de Germain *et al.* [5] el autor utiliza la distribución Γ para ajustar estos datos. En el trabajo recientemente presentado, [6] se prueba que para imágenes SAR polarimétricas la distribución \mathcal{G} polarimétrica tiene un muy buen comportamiento.

Para poder aplicar el algoritmo descrito en la sección 3 es necesario elegir una distribución adecuada para los datos con los que se está trabajando, que corresponden a datos SAR filtrados con el filtro *stack*. Para elegir la distribución, realizamos el siguiente experimento.

Se toman 100 imágenes sintéticas de 128×64 cada una generada con la distribución \mathcal{G}_A^0 con parámetros $\alpha_1 = -1,5$, $\gamma_1 = 1$. Se procesa cada una de las imágenes con sus respectivos filtros obtenidos en la fase de entrenamiento, como se explica en la sección 2, utilizando la media de los datos como imagen a obtener (imagen ideal). Observamos que después de filtrar las imágenes, el skewness de los datos se reduce, acercándose a cero, lo que indica que la distribución de los niveles de gris de la imagen, después de ser filtrada se acerca a un modelo gaussiano (ver apéndice A). Por esa razón, se utiliza en este trabajo, la distribución normal para modelar los datos de la imagen filtrada.

La Figura 2 muestra el tipo de imágenes que se utilizaron para evaluar los datos y elegir la distribución. La Figura 2(a) muestra una imagen sintética original generada con el modelo estadístico \mathcal{G}_A^0 . La Figura 2(b) muestra el nivel de gris correspondiente a la media de los datos. La Figura 2(c) muestra la imagen filtrada.

El Cuadro 1 muestra los valores del skewness para 10 de las imágenes utilizadas. La primer fila corresponde al skewness de la imagen sin filtrar y la segunda al de la imagen filtrada. Se observa que se reduce el valor en la segunda fila, acercándose a cero.

Skewness Original	0.84	0.83	0.81	0.86	0.85	0.88	0.85	0.90	0.86	0.93
Skewness Filtrada	0.16	0.20	0.18	0.13	0.23	0.13	0.13	0.17	0.15	0.16

Cuadro 1: Skewness calculado para 10 de las imágenes filtradas.

Los resultados de aplicar el algoritmo completo a imágenes SAR reales se muestran en la sección 5.

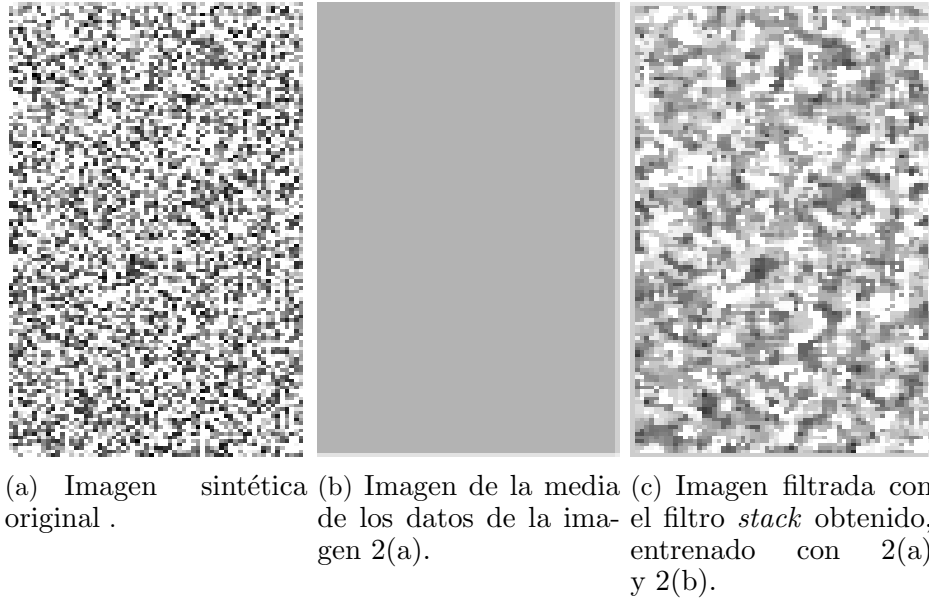


Figura 2: Ejemplo de las imágenes sintéticas generadas para entrenar el filtro.

5. Resultados

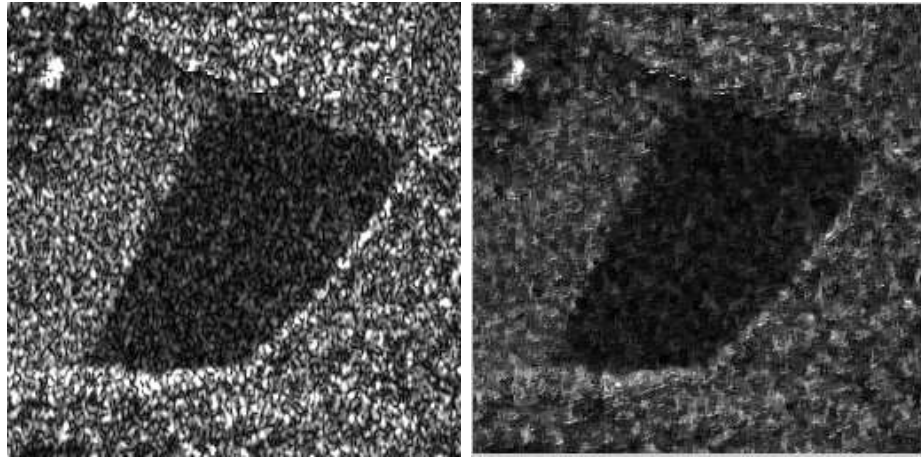
En esta sección se presentan los resultados obtenidos al aplicar el algoritmo a una imagen SAR real. La Figura 3(a) corresponde a la imagen SAR original. La Figura 3(b) corresponde a la imagen filtrada usando la media de los datos de entrada para cada región. La Figura 3(c) es el resultado de aplicar el algoritmo de detección de contornos descrito en la sección 3.

6. Conclusiones y Trabajo Futuro

En este trabajo se presenta un método para detectar el contorno de regiones en imágenes con ruido *speckle*. En primer lugar se aplica a la imagen un filtro *stack* que suaviza el ruido *speckle*. Este filtro se entrena usando la media de las diferentes regiones. Para encontrar el contorno se aplica un algoritmo de detección de borde, basado en curvas b-spline deformables. Este proceso es eficiente en la extracción de contornos de regiones. En este momento estamos trabajando en la utilización de otras medidas estadísticas, como la kurtosis, para demostrar que la distribución normal ajusta a los datos filtrados. El siguiente paso es poder aplicar el filtro dentro de una región de interés, de forma tal que el algoritmo de filtrado no tenga que analizar toda la imagen, sino aquellas áreas donde se encuentran los bordes.

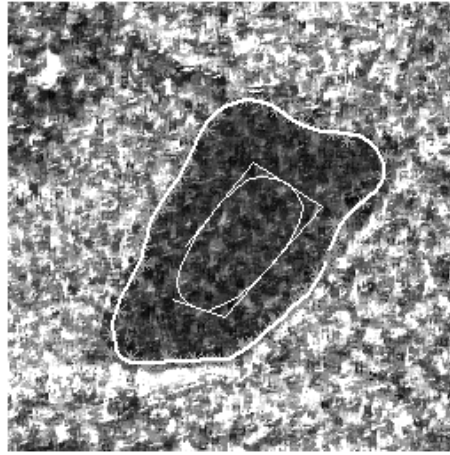
A. Apéndice: Medida de Skewness

Una tarea importante en muchos análisis estadísticos es caracterizar la ubicación y variabilidad de un conjunto de datos. Un estudio en este sentido, incluye el cálculo y análisis de skewness.



(a) Imagen SAR real.

(b) Imagen filtrada entrenando con la media de los datos de entrada.



(c) Detección del borde entre las diferentes regiones

Figura 3: Resultado de aplicar el algoritmo para detección de bordes entre diferentes regiones.

Skewness es una medida de simetría del conjunto.

Para datos univariados Y_1, Y_2, \dots, Y_N , la fórmula de skewness es:

$$SK = \frac{\sum_{i=1}^N (Y_i - \bar{Y})^3}{(N-1)s^3}$$

donde \bar{Y} es la media, s es la desviación standar, y N es el tamaño de la muestra. El skewness SK para una distribución normal es cero y cualquier conjunto de datos simétrico, debería tener skewness cerca de cero.

Referencias

- [1] J. Astola and P. Kuosmanen. *Fundamentals of Nonlinear Digital Filtering*. CRC Press, Boca Raton, 1997.
- [2] A. Blake and M. Isard. *Active Contours*. Springer Verlag, 1998.

- [3] M. J. Gambini, M. Mejail, H. Jacobo-Berlles, and A. C. Frery. Feature extraction in speckled imagery with b-spline deformable contours. In *European Conference on Circuit Theory and Design*, Cork, Ireland, 2005. IEEE.
- [4] M. J. Gambini, M. Mejail, J. C. Jacobo-Berlles, H. Muller, and A. C. Frery. Automatic contour detection in sar images. In *5th European Conference on Synthetic Aperture Radar ISBN = 3-8007-2828-1 198*, pages 993–996, 2004.
- [5] O. Germain and P. Réfrégier. Edge location in sar images: Performance of the likelihood ratio filter and accuracy improvement with an active contour approach. *IEEE Transaction on Image Processing*, 10(1), 2001.
- [6] J. Jacobo Berlles. *Nuevas Familias de Distribuciones Polarimétricas para Imágenes SAR*. PhD thesis, Universidad de Buenos Aires, 2005.
- [7] M. E. Mejail, A. C. Frery, J. Jacobo-Berlles, and O. H. Bustos. Approximation of distributions for SAR images: Proposal, evaluation and practical consequences. *Latin American Applied Research*, 31:83–92, 2001.
- [8] D. F. Rogers and J. A. Adams. *Mathematical Elements for Computer Graphics*. McGraw-Hill, New York, USA, 2 edition, 1990.
- [9] Jisang Yoo, Kelvin L. Fong, Jr-Jen Huang, Edward J. Coyle, and George B. Adams III. A fast algorithm for designing stack filters. *IEEE Trans.on image processing*, 8(8):772–781, 8 1999.